**LAPORAN KELOMPOK**

**“STRUKTUR DATA”**

**D4 MANAJEMEN INFORMATIKA**



ALVIN NOOR HIDAYAH

21091397016

ACHMAD NURS SYURURI ARIFIN

21091397024

MUHAMMAD DHAFA JAWADIL UBAID

21091397058

MUHAMMAD AL FAIZ PUTRA J.

21091397072

TEGUH ALDIANTO  
21091397076

1. **Codingan insertion sort**

#include <iostream>

using namespace std;

int main(){

int y;

cout<<"masukan banyak array:";

cin>>y;

int x[y];

// disini 5-8 itu inputan dari user dimana variabel y banyak data//

// dan array X wadah untuk data yang akan diinputkan//

for(int i=0; i<y; i++){

cout<<"masukan angka ke "<<i<<" :";

//disini saya melakukan perulangan untuk mengisi index

// dari setiap array mulai dari array ke 0//

cin>>x[i];

cout<<endl;

}

for(int i=1; i<y; i++){

int key = x[i];

int j = i-1;

while(j>=0 && x[j] > key){

x[j+1] = x[j];

j--;

//disini pertama akan mulai perulangan

//dari indeks ke 1 yaitu variabel i

//variabel i akan menajdi kunci untuk dibandingkan sebelumnya

//jika bilangan sebelumnya lebih besar, maka key akan kembali

//key kembali dibandingkan dengan bil. ke 2 hingga menemukan

// sampai key menemukan bilangan yang diinginkan

//untuk menghasilkan nilai shortingan yang diinginkan disini nilai harus berada disebelah paling kanan//

//yaitu harus di variabel key tersebut//

}

x[j+1] = key;

cout<<"proses sorting"<<endl;

//disni akan terjadi proses sort yang akan menghitung angka//

//kemudian melakukan proses sorting angka yang diinginkan//

for(int m=0;m<y;m++){

cout<<x[m]<<" ";

}

cout<<endl;

}

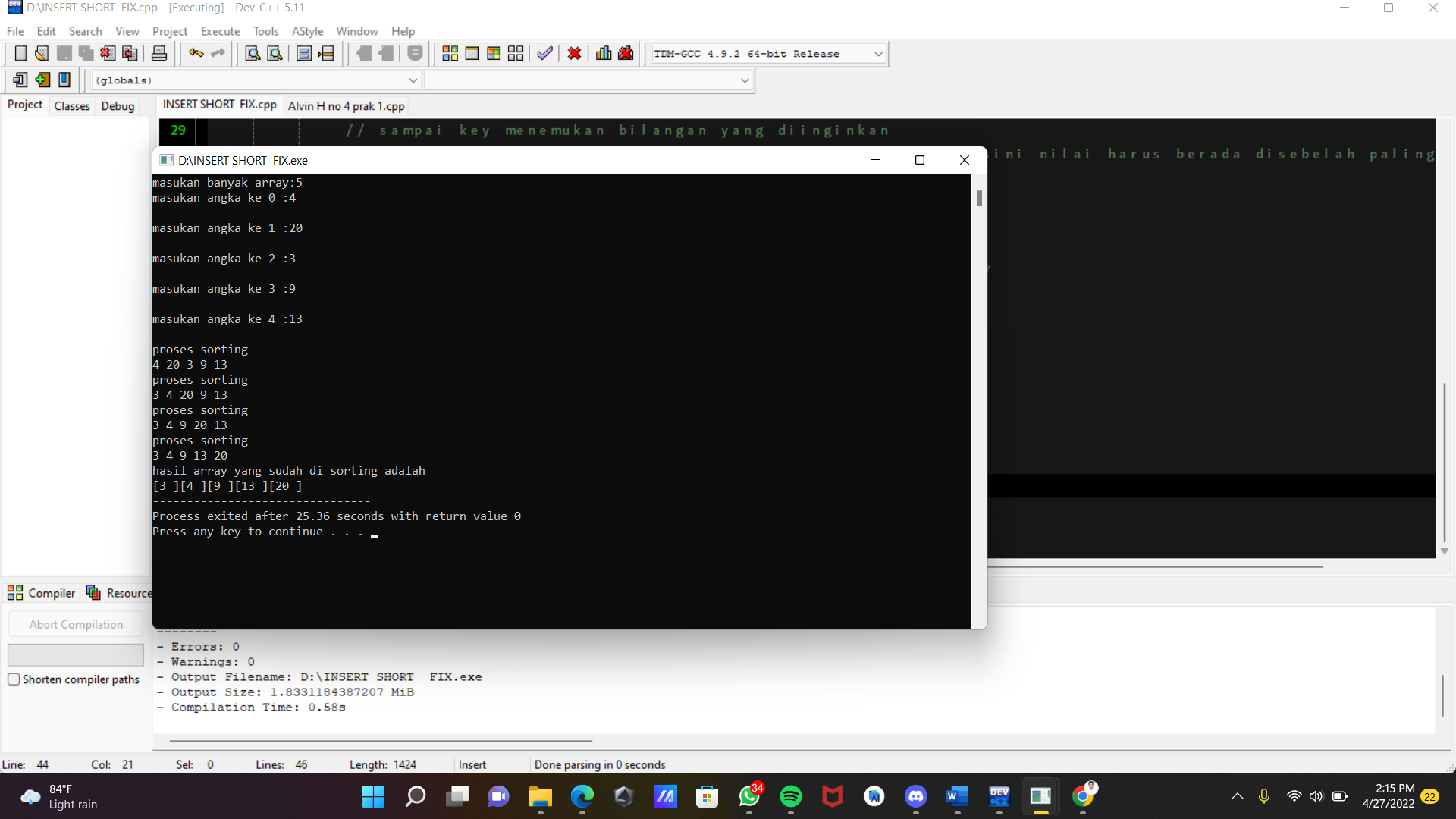
cout<<"hasil akhir"<<endl;

for(int m=0;m<y;m++){

cout<<x[m]<<" ";

}

}



b. **codingan selection sort**

#include<iostream>

using namespace std;

void swapping(int &a, int &b){ //menukar isi dari a dan b

int temp;

temp = a;

a = b;

b = temp;

}

void display(int \*array, int size){

for(int i = 0; i<size; i++)

cout << array[i] << " ";

cout << endl;

}

void selectionSort(int \*array, int size){

int i, j, imin;

for(i = 0; i<size-1; i++){

imin = i; //mendapatkan indeks data minimum

for(j = i+1; j<size; j++)

if(array[j] < array[imin])

imin = j;

//menempatkan di posisi yang benar

swap(array[i], array[imin]);

}

}

int main(){

int n;

cout << "Masukkan jumlah elemen:";

cin >> n;

int arr[n]; //membuat array dengan jumlah elemen yang diberikan

cout << "Masukkan elemen:" << endl;

for(int i = 0; i<n; i++){

cin >> arr[i];

}

cout << "Array sebelum sorting:";

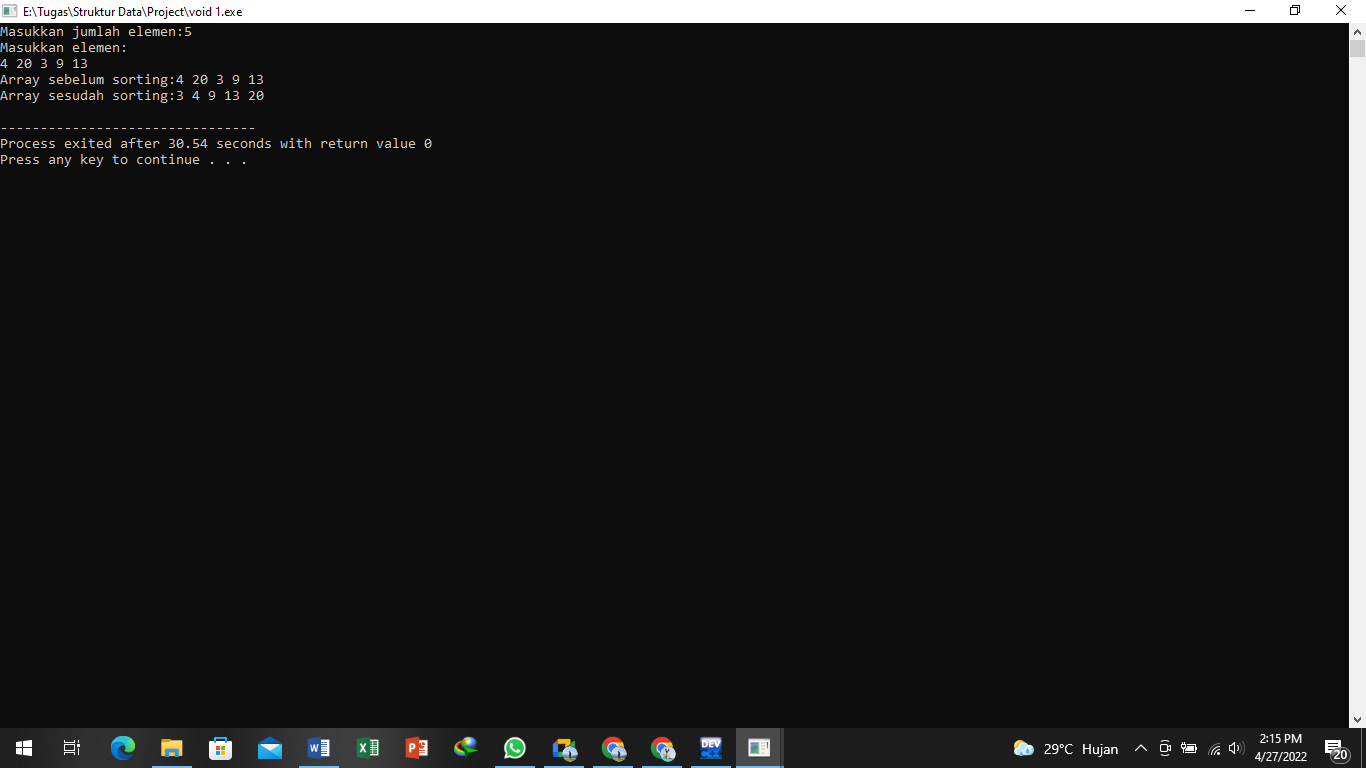
display(arr, n);

selectionSort(arr, n);

cout << "Array sesudah sorting:";

display(arr, n);

}



c. **codingan merge sort**

#include <iostream>

using namespace std;

void merge(int arr[], int l, int m, int r)

{

int i, j, k;

int n1 = m - l + 1;

int n2 = r - m;

int L[n1], R[n2];

for (i = 0; i < n1; i++)

L[i] = arr[l + i];

for (j = 0; j < n2; j++)

R[j] = arr[m + 1 + j];

i = 0;

j = 0;

k = l;

while (i < n1 && j < n2)

{

if (L[i] <= R[j])

{

arr[k] = L[i];

i++;

}

else

{

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1)

{

arr[k] = L[i];

i++;

k++;

}

while (j < n2)

{

arr[k] = R[j];

j++;

k++;

}

}

void mergeSort(int arr[], int l, int r)

{

if (l < r)

{

int m = l + (r - l) / 2;

mergeSort(arr, l, m);

mergeSort(arr, m + 1, r);

merge(arr, l, m, r);

}

}

void show(int A[], int size)

{

int i;

for (i = 0; i < size; i++)

cout << A[i] << " ";

}

int main()

{

int size;

cout << "\nMasukan Banyak Data : ";

cin >> size;

int arr[size];

for (int i = 0; i < size; ++i)

{

cout << "\nMasukan Data array ke "<<i<<" :";

cin >> arr[i];

}

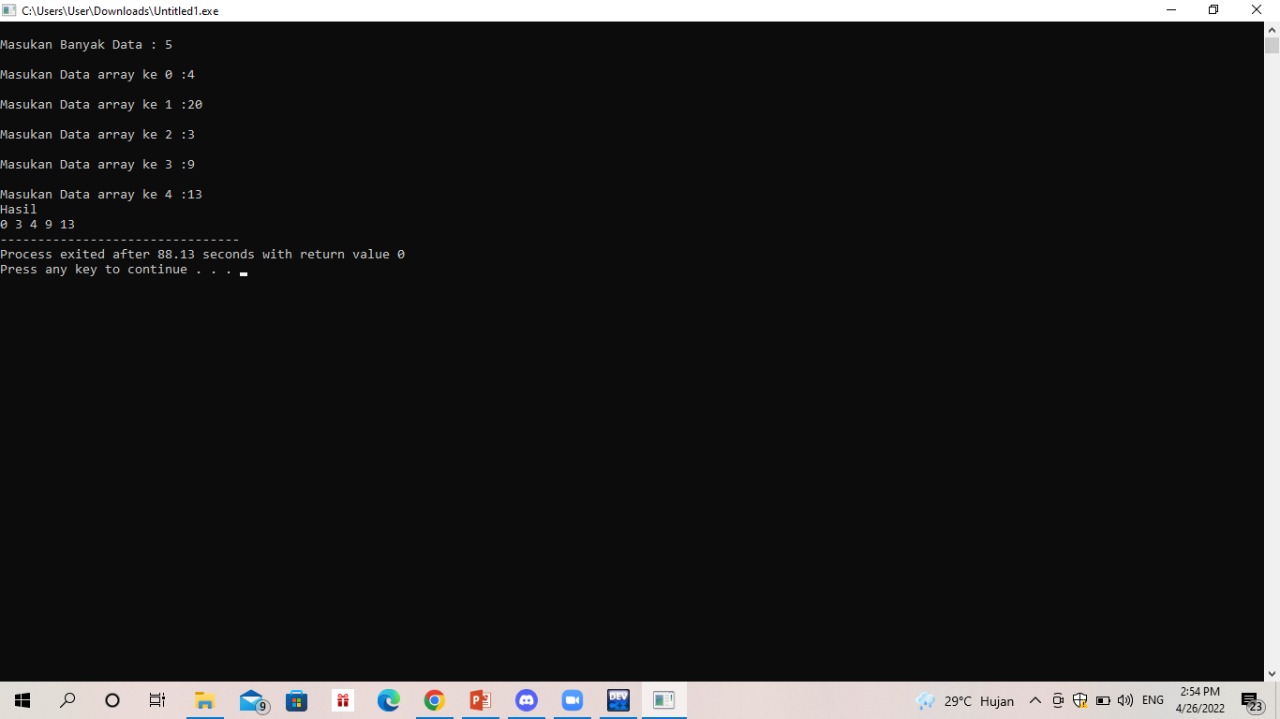
mergeSort(arr, 0, size);

cout << "Hasil\n";

show(arr, size);

return 0;

}



d. **codingan buble sort**

#include <iostream>

#include <conio.h>

using namespace std;

main()

{

int data [100];

int i, j, k, tmp;

cout<<"Pengurutan Bilangan Buble Sort \n\n";

cout<<"Masukkan Jumlah Bilangan : ";cin>>k;

for(i=0; i<k; i++)

{

cout<<"Masukkan Angka Ke "<<(i+1)<<" : ";

cin>>data[1];

}

cout<<"\nAngka Sebelum Diurutkan : "<<endl;

for(i=0; i<k; i++)

{

cout<<data[i]<<" ";

}

cout<<endl;

for(i=0; i<k; i++)

{

for(j=i+1; j<k; j++)

{

if(data[i]>data[j])

{

tmp=data[i];

data[i]=data[j];

data[j]=tmp;

}

}

}

cout<<"\nAngka Setelah Diurutkan : "<<endl;

for(i=0; i<k; i++)

{

{

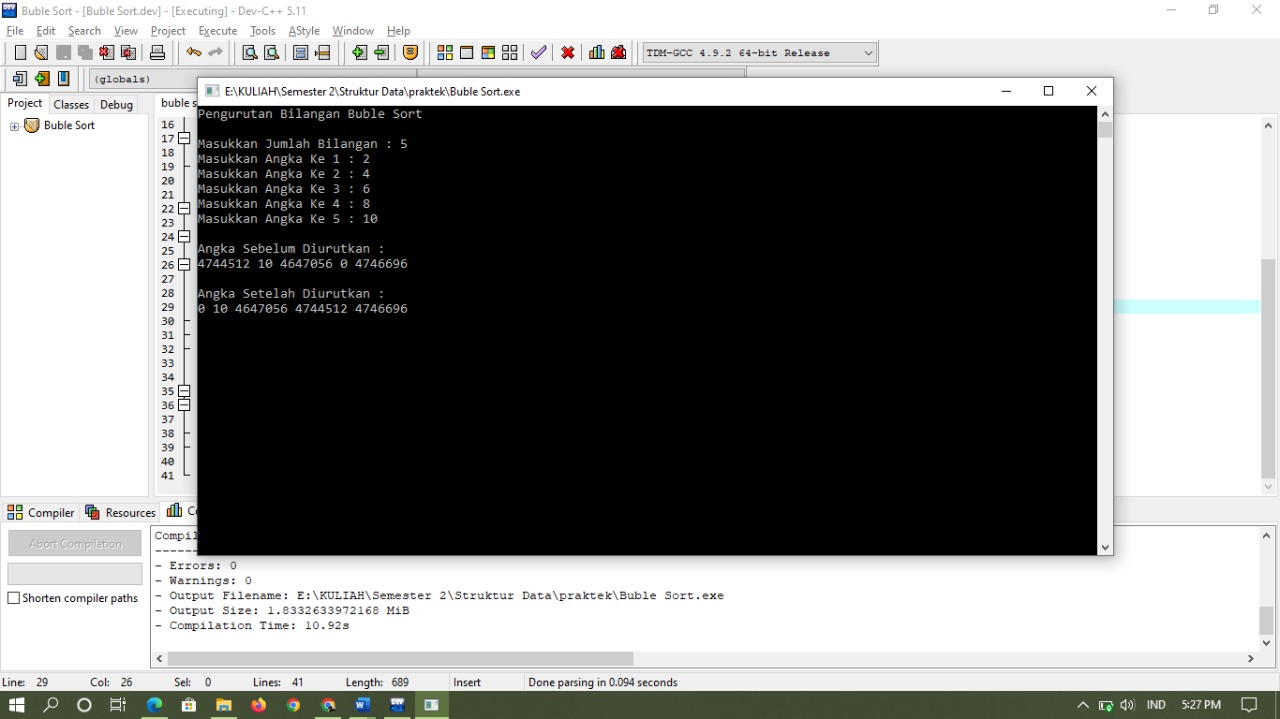
cout<<data[i]<<" ";

}

}

getch();

}



e. **Codingan Shell Sort**

#include<iostream>

using namespace std;

// sebuah fungsi yang mengimplementasikan shellshort.

void ShellSort(int a[], int n)

{

int i, j, k, temp;

// celah 'i' antara indeks elemen yang akan dibandingkan awalnya n/2.

for(i = n/2; i > 0; i = i/2)

{

for(j = i; j < n; j++)

{

for(k = j-i; k >= 0; k = k-i)

{

// jika nilai pada indeks yang lebih tinggi lebih besar, maka putuskan loop.

if(a[k+i] >= a[k])

break;

// ganti nilai sebaliknya.

else

{

temp = a[k];

a[k] = a[k+i];

a[k+i] = temp;

}

}

}

}

}

int main()

{

int n, i;

cout<<"\nmasukkan jumlah elemen data yang akan diurutkan: ";

cin>>n;

int arr[n];

for(i = 0; i < n; i++)

{

cout<<"masukkan elemen "<<i+1<<": ";

cin>>arr[i];

}

ShellSort(arr, n);

// mencetak data yang diurutkan.

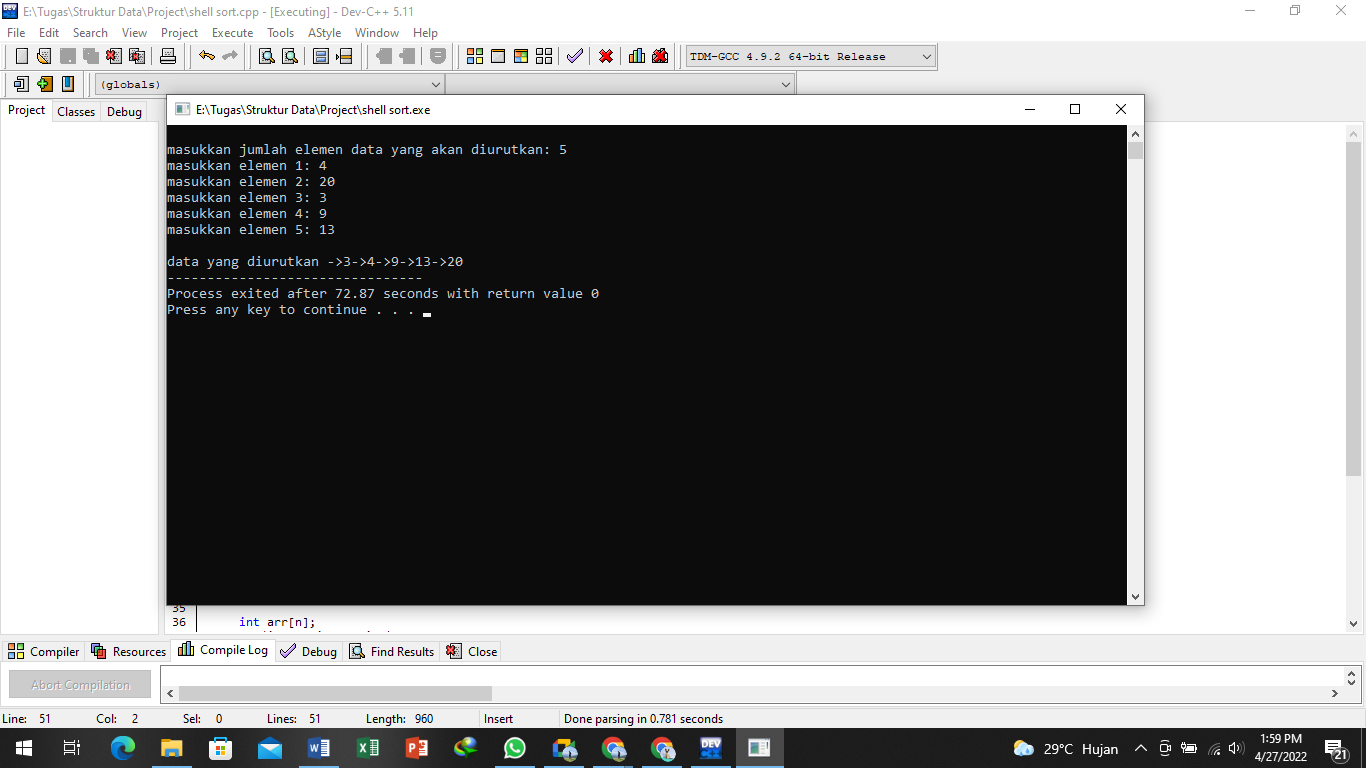
cout<<"\ndata yang diurutkan ";

for (i = 0; i < n; i++)

cout<<"->"<<arr[i];

return 0;

}



**KESIMPULAN**

Berdasarkan hasil output dari kodingan diatas maka dapat disimpulkan bahwa sortingan paling cepat adalah Insertion Sort dan sortingan paling lama adalah Bubble Sort.